

CLAIMS

1. A method for generating a tamper resistant version of a software program including a stream of data blocks, comprising:
- undertaking a predetermined number of iterations of forward plain text chaining of the blocks followed by backward plain text chaining of the blocks.
2. The method of claim 1, further comprising XORing a first block with an adjacent block to render a chained block.
3. The method of Claim 2, further comprising scrambling chained blocks using a cipher.
4. The method of Claim 3, comprising scrambling a chained block using at least one but not all rounds of the cipher to render a scrambled block before chaining the chained block to another block.
5. The method of Claim 4, comprising descrambling the chained block using only a single round of the cipher to render a result and then XORing the result with an adjacent block.
6. A computer program device, comprising:

2 a computer program storage device including a program of instructions
3 usable by an encryption computer, comprising:

4 logic means for chaining a data block to a plain text version of an
5 adjacent block in the stream to render a chained block;

6 logic means for scrambling the chained block using a first round of a
7 cipher to render a scrambled block; and

8 logic means for iterating the means for scrambling and chaining using
9 subsequent rounds of the cipher.

1 7. The computer program device of Claim 6, wherein the means for
2 iterating iterates forward and backward through the stream, using successive rounds
3 of the cipher.

4 8. A computer system for encrypting a stream of data blocks, comprising
5 a processor programmed to execute method acts including:

6 (a) receiving a sequence of N blocks;

7 (b) initializing a previous block variable B;

8 (c) for i=1 to N, executing a DO loop comprising:

(c)(1) XORing an ith block with B to render a modified ith
block;

(c)(2) setting B equal to the modified ith block;

9
10
11
12
13
14
15
16
17
18
19
20
21
22

Q2

(c)(3) scrambling the modified ith block using at least one
round of a cipher;
(c)(4) incrementing "i" by unity and returning to act (b)(1);
(d) initializing a previous block variable B;
(e) for i=N to 1, executing a DO loop comprising:
(e)(1) XORing an ith block with B, yielding a modified ith
block;
(e)(2) setting B to the modified ith block;
(e)(3) scrambling the modified ith block using at least one next
round of a cipher;
(e)(4) decrementing "i" by unity and returning to act (b)(1); and
(f) determining whether a predetermined number of iterations
have been executed, and if not, returning to act (b) using a next round
of the cipher, otherwise outputting an encrypted stream of data blocks.

9. The computer system of Claim 8, wherein the stream of data blocks is
established by a computer program.

10. The computer system of Claim 9, wherein a respective round of the
cipher is used for each iteration.

1 11. A method for generating a tamper resistant version of a software
2 program including a stream of data blocks, comprising:

3 providing a cipher defining rounds;

4 iterating through the rounds of the cipher by iterating through
5 respective outer loops of forward plain text chaining followed by backward
6 plain text chaining; and

7 during each forward portion of an outer loop, applying a respective
8 round of the cipher to each block, and during each backward portion of an
9 outer loop, applying a respective round of the cipher to each block.

10 12. The method of Claim 11, further comprising:

11 (a) receiving a sequence of N blocks;

12 (b) initializing a previous block variable B;

(c) for i=1 to N, executing a DO loop comprising:

(c)(1) XORing an ith block with B to render a modified ith
block;

(c)(2) setting B equal to the modified ith block;

(c)(3) scrambling the modified ith block using at least one
round of a cipher;

(c)(4) incrementing "i" by unity and returning to act (b)(1);

(d) initializing a previous block variable B;

(e) for i=N to 1, executing a DO loop comprising:

13 (e)(1) XORing an ith block with B, yielding a modified ith
14 block;
15 (e)(2) setting B to the modified ith block;
16 (e)(3) scrambling the modified ith block using at least one next
17 round of a cipher;
18 (e)(4) decrementing "i" by unity and returning to act (b)(1); and
19 (f) determining whether a predetermined number of iterations
20 have been executed, and if not, returning to act (b) using a next round
21 of the cipher, otherwise outputting an encrypted stream of data blocks.

13. A method for generating a tamper resistant version of a software
program including a stream of data blocks, comprising:

scrambling a block using one and only one round of a cipher; then
chaining the block to another block to render a chained block; then
scrambling the chained block using one and only round of the cipher.

14. The method of Claim 13, further comprising:

(a) receiving a sequence of N blocks;
(b) initializing a previous block variable B;
(c) for i=1 to N, executing a DO loop comprising:
(c)(1) XORing an ith block with B to render a modified ith
block;

7 (c)(2) setting B equal to the modified ith block;
8 (c)(3) scrambling the modified ith block using at least one
9 round of a cipher;
10 (c)(4) incrementing "i" by unity and returning to act (b)(1);
11 (d) initializing a previous block variable B;
12 (e) for i=N to 1, executing a DO loop comprising:
13 (e)(1) XORing an ith block with B, yielding a modified ith
14 block;
15 (e)(2) setting B to the modified ith block;
16 (e)(3) scrambling the modified ith block using at least one next
17 round of a cipher;
18 (e)(4) decrementing "i" by unity and returning to act (b)(1); and
19 (f) determining whether a predetermined number of iterations
20 have been executed, and if not, returning to act (b) using a next round
21 of the cipher, otherwise outputting an encrypted stream of data blocks.

1 15. A computer system for decrypting a stream of data blocks, comprising
2 a processor programmed to execute method acts including:

- 3 (a) receiving a sequence of N blocks;
4 (b) for i= N to 1, executing a DO loop comprising:
5 (b)(1) reverse XORing an ith block with a block_{i-1};

6 (b)(2) unscrambling the i^{th} block using a round of a cipher to
7 render an unscrambled block;

8 (b)(3) determining whether a block_{i-1} exists, and if not,
9 proceeding to act (c), otherwise;

10 (b)(4) decrementing "i" by unity and returning to act (b)(1);

11 (c) for $i = 1$ to N , executing a DO loop comprising:

12 (c)(1) reverse XORing an i^{th} block with a block_{i+1} ;

13 (c)(2) unscrambling the i^{th} block using a single round of a
14 cipher to render an unscrambled block;

15 (c)(3) determining whether a block_{i+1} exists, and if not,
16 proceeding to act (d), otherwise;

17 (c)(4) incrementing "i" by unity and returning to act (c)(1);

18 (d) determining whether a predetermined number of iterations have
19 been executed, and if not, returning to act (b) using a next round of the cipher,
20 otherwise outputting a decrypted stream of data blocks.

1 16. The computer system of Claim 15, wherein the stream of data blocks
2 is established by a computer program.

1 17. The computer system of Claim 16, wherein a respective round of the
2 cipher is used for each iteration.